

Computación Paralela y Distribuida

2021-I

José Fiestas

28/05/21

Universidad de Ingeniería y Tecnología
jfiestas@utec.edu.pe

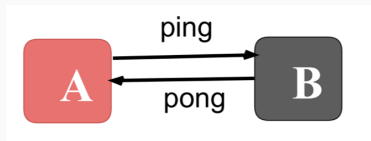
Práctica Dirigida 04:

MPI

Puntaje: 4 pts.

Ejercicio 1: Ping Pong (1 pt)

Programe el algoritmo de Ping-Pong entre dos procesos A y B tal que se intercambie un array de floats de tamaño variable. Duplicar la dimensión en cada iteración desde 2 a 2^{18} . Mida adecuadamente los tiempos de ejecución utilizando `MPI_Wtime()`. No permita la ejecución del código si $p > 2$



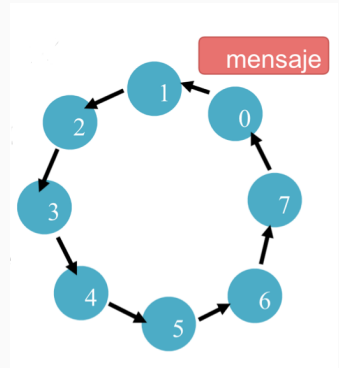
Ejercicio 2: Latencia en Anillo (1 pt)

Desarrolle un código en MPI que mida la latencia de un anillo de p procesos, entendiendo por latencia el tiempo que tarda un mensaje de tamaño 0 en circular entre todos los procesos.

En el anillo

P_0 envía el mensaje a P_1 , que lo recibe y reenvía a P_2 , y así sucesivamente hasta que llega a P_{p-1} , que lo enviará a P_0 .

Para tener un resultado más preciso, realice varios experimentos y promedie resultados



Utilice el modelo PRAM desarrollado en la tarea 02

Ejercicio 3: Multiplicación matriz-vector (1 pts)

El siguiente fragmento de código calcula el producto de una matriz cuadrada por un vector, ambos de la misma dimensión N :

```
int i, j;
    int A[N][N], v[N], x[N];
    leer(A,v);
    for (i=0;i<N;i++) {
        x[i]=0;
        for (j=0;j<N;j++) x[i] += A[i][j]*v[j];
    }
    escribir(x);
```

Ejercicio 3: (cont.)

- a) Programe en MPI el producto en paralelo, teniendo en cuenta que el proceso P_0 obtiene inicialmente la matriz A y el vector v , realiza una distribución de A por bloques de filas consecutivas sobre todos los procesos y envía v a todos. P_0 debe obtener el resultado.
- b) Obtener el costo computacional y de comunicaciones del algoritmo paralelo y comparar los resultados con la complejidad teórica esperada. Calcular el speed-up y la eficiencia

Ejercicio 4: norma-infinito (1 pt)

Paralelice la norma-infinito de una matriz, definida como

$|x|_{\infty} = \max_{i=1}^n \{ \sum_{j=1}^m |x_{ij}| \}$, como lo muestra el código secuencial

```
#include <math.h>
#define N 800
double infNorm(double A[][N]) {
    int i,j;
    double s, norm;
    norm=fabs(A[0][0]);
    for(int i=0; i<N; i++) {
        s=0.0;
        for (int j=0; j<N; j++)
            s+=fabs(A[i][j]);
        if(s>norm)
            norm=s; }
    return norm;
}
```

Ejercicio 4 (cont.)

- a) Utilice comunicación colectiva en MPI. Asumir que el tamaño del problema es un múltiplo exacto del número de procesos.
- b) Obtener el costo computacional y de comunicaciones del algoritmo paralelo. Asumir que la operación `fabs` tiene un costo despreciable, así como las comparaciones.
- c) Calcular el speed-up y la eficiencia si el tamaño del problema tiende a infinito